

PATENT ABSTRACTS OF JAPAN

(11)Publication number : 2002-055936

(43)Date of publication of application : 20.02.2002

(51)Int.CI.

G06F 13/12
G06F 13/14
G06F 13/38

(21)Application number : 2000-239645

(71)Applicant : TOSHIBA CORP

(22)Date of filing : 08.08.2000

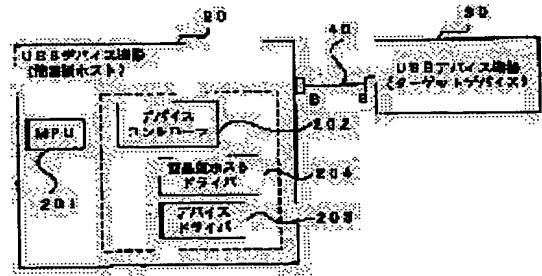
(72)Inventor : NAMIHIRA SHINJI

(54) USB DEVICE, AND SYSTEM AND METHOD FOR COMMUNICATION BETWEEN USB DEVICES

(57)Abstract:

PROBLEM TO BE SOLVED: To provide a USB device, and a system and a method for communications between USB devices which enable the USB devices to directly transfer data by themselves without requiring any USB host device.

SOLUTION: This USB device 20 has at least an MPU 201, a USB device controller 202 and a USB device driver 203, and is provided with a simple host driver 204 which has a process for recognizing the other USB device 30 connected by a connection cable 40 and a process for transferring data between the USB devices. The simple host driver 204 enables the direct data transfer with the other USB device 30.



LEGAL STATUS

[Date of request for examination]

[Date of sending the examiner's decision of rejection]

[Kind of final disposal of application other than the examiner's decision of rejection or application converted registration]

[Date of final disposal for application]

[Patent number]

[Date of registration]

[Number of appeal against examiner's decision of rejection]

[Date of requesting appeal against examiner's decision of rejection]

[Date of extinction of right]

Copyright (C); 1998,2003 Japan Patent Office

(19)日本国特許庁 (JP)

(12) 公開特許公報 (A)

(11)特許出願公開番号

特開2002-55936

(P2002-55936A)

(43)公開日 平成14年2月20日 (2002.2.20)

(51)Int.Cl.
G 0 6 F 13/12
13/14
13/38

識別記号
3 1 0
3 3 0
3 5 0

F I
G 0 6 F 13/12
13/14
13/38

テーマコード(参考)
3 1 0 Z 5 B 0 1 4
3 3 0 C 5 B 0 7 7
3 5 0

審査請求 未請求 請求項の数10 O L (全 10 頁)

(21)出願番号 特願2000-239645(P2000-239645)

(22)出願日 平成12年8月8日 (2000.8.8)

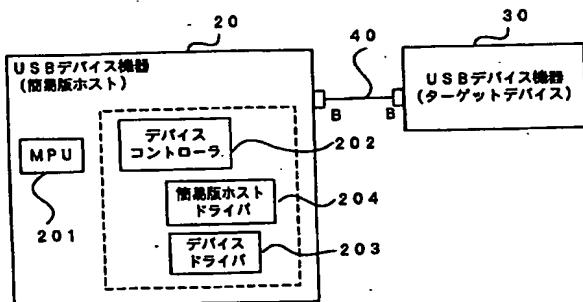
(71)出願人 000003078
株式会社東芝
東京都港区芝浦一丁目1番1号
(72)発明者 波平 真二
東京都青梅市末広町2丁目9番地 株式会
社東芝青梅工場内
(74)代理人 100083161
弁理士 外川 英明
Fターム(参考) 5B014 EB01 FA05 FA11 FB03 FB04
GD05 GD22 GD23 GD33 GE05
HC06 HC08
5B077 AA18 BB06 NN02

(54)【発明の名称】 USBデバイス機器、USBデバイス機器間の通信システムおよび通信方法

(57)【要約】

【課題】 USBホスト機器を必要とせずに、USBデバイス機器同士のみでの直接データ転送を可能にするUSBデバイス機器、USBデバイス機器間の通信システム、および通信方法を提供する。

【解決手段】 少なくともMPU201と、USBデバイスコントローラ202と、USBデバイスドライバ203とを有するUSBデバイス機器20であって、接続ケーブル40によって接続される他のUSBデバイス機器30を認識する処理とUSBデバイス間データ転送を実行する処理を持った簡易版ホストドライバ204を設け、前記簡易版ホストドライバ204によって前記他のUSBデバイス機器30との直接データ転送を可能にした。



【特許請求の範囲】

【請求項1】 少なくともMPUと、USBデバイスコントローラと、USBデバイスドライバとを有するUSBデバイス機器であって、接続ケーブルによって接続される他のUSBデバイス機器を認識する処理とUSBデバイス間データ転送を実行する処理を持った簡易版ホストドライバを設け、前記簡易版ホストドライバによって前記他のUSBデバイス機器との直接データ転送を可能にしたことを特徴とするUSBデバイス機器。

【請求項2】 前記他のUSBデバイス機器との接続を検知する手段を具備し、前記検知手段によって前記他のUSBデバイス機器との接続が行なわれたことを検出した後、前記接続ケーブルの電源ラインにUSBバス電源を供給して前記簡易版ホストドライバをスタートさせることを特徴とする請求項1項記載のUSBデバイス機器。

【請求項3】 前記接続ケーブルの両端コネクタは、Bタイプコネクタを使用することを特徴とする請求項2記載のUSBデバイス機器。

【請求項4】 前記簡易版ホストドライバは、前記他のUSBデバイス機器に対し標準デバイスリクエストを必要なだけ発行して、前記他のUSBデバイス機器のデバイス情報、コンフィグレーション情報を取得し、前記他のUSBデバイスに通信情報をセットすることを特徴とする請求項1項記載のUSBデバイス機器。

【請求項5】 前記簡易版ホストドライバの構成要素として、デバイスアドレスの割り付けるためのリクエスト処理の際に、前記他のUSBデバイス機器にセットするデバイスアドレス値を予め定められた固定値にセットしておくことを特徴とする請求項1記載のUSBデバイス機器。

【請求項6】 前記簡易版ホストドライバの構成要素として、通信エンドポイント0の通信パケットサイズ、および最大バス電力消費量を予め定められた固定値にしておくことを特徴とする請求項1記載のUSBデバイス機器。

【請求項7】 前記簡易版ホストドライバは、USBホスト機能であるトークンパケットを発行できることを特徴とする請求項1記載のUSBデバイス機器。

【請求項8】 少なくともMPUと、USBデバイスコントローラと、USBデバイスドライバとを有する第1および第2のUSBデバイス機器と、前記第1および第2のUSBデバイス機器を接続する両端Bタイプコネクタを持つケーブルと、

前記第1のUSBデバイス機器に設けられた、前記第2のUSBデバイス機器を認識する処理とUSBデバイス間データ転送を実行する処理を持った簡易版ホストドライバとを有し、

前記第1のUSBデバイス機器と前記第2のUSBデバイス機器との間で直接データ転送を可能にしたことを特

徴とするUSBデバイス機器間の通信システム。

【請求項9】 少なくともMPUと、USBデバイスコントローラと、USBデバイスドライバとを有する第1および第2のUSBデバイス機器間の通信方法であつて、

前記第1のUSBデバイス機器に設けられた、前記第2のUSBデバイス機器を認識する処理とUSBデバイス間データ転送を実行する処理を持った簡易版ホストドライバによって、

前記他のUSBデバイス機器に対し標準デバイスリクエストを必要なだけ発行して、前記他のUSBデバイス機器のデバイス情報、コンフィグレーション情報を取得し、前記他のUSBデバイスに通信情報をセットし、前記第1のUSBデバイス機器から前記第2のUSBデバイス機器へトークンパケットを発行してデータ通信を行なうことを特徴とするUSBデバイス機器間の通信方法。

【請求項10】 少なくともMPUと、USBホストコントローラと、USBホストドライバとを有する第1および第2のUSBホスト機器と、前記第1および第2のUSBホスト機器を接続する両端Aタイプコネクタを持つケーブルと、前記第1のUSBホスト機器に設けら、前記第2のUSBホスト機器との間のデータ転送を実行する処理を持った簡易版デバイスドライバとを有し、前記第1のUSBホスト機器と前記第2のUSBホスト機器との間で直接データ転送を可能にしたことを特徴とするUSBホスト機器間の通信システム。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】 本発明は、USBホスト機器を接続しないで、USBデバイス機器の間のデータ通信を可能にしたUSBデバイス機器、USBデバイス機器間の通信システム、および通信方法に関する。

【0002】

【従来の技術】 従来のUSB機器間の通信システムにおいては、通信を取り纏めるUSBホスト機能を持ったUSB機器の存在は不可欠であった。このUSB機器のUSBホスト機能には、USBホストに接続された複数のUSBデバイスを認識し、デバイスアドレスを割り振って管理する機能（接続されたUSBデバイスを認識する処理をエニュメレーションと呼ぶ）、およびUSBホスト機器とUSBデバイス機器間、またはUSBデバイス機器とUSBデバイス機器間の全てのUSB通信を仲介／取り締める機能（USBデータ転送を開始するトークンパケットの発行、通信エラーチェック等）などが挙げられる。

【0003】 図11に、従来のUSB機器の通信システムの接続構成を示す。この例では、USBホスト機器（PC）と、USBデバイス機器として、例えばデ

ジタルスチルカメラ11、プリンタ12が接続されている様子を示す。この図11に示すように、一般的にUSBホスト機能を持ったUSBホスト機器10として、PC(パソコン用コンピュータ)である場合が多い。しかしながら、USBデバイス機器11と12との間でのデータ通信を行う際にも(この例では、カメラ11からの画像出力データをプリンタ12にて印刷する場合のデータ通信)、USBホスト機器10を介して行なう必要があり、大変不便であった。そこで、USBホスト機器10の介在しないで、USBデバイス機器同士の間で直接データ通信を実現するシステムが求められる。

【0004】その一つとして、USBホスト機能をそのままUSBデバイス機器へ組み込んでしまうという手法がある。USBホスト機能を組み込まれたUSBデバイス機器は、USBデバイス機器としての機能だけでなく、USBホスト機器としての機能も備えることになる。その為、PCなどのUSBホスト機器が接続された場合には、従来通りUSBデバイスとして機能するが、通信ターゲットとなる別のUSBデバイス機器が接続された場合には、USBホスト機器として機能してUSB通信を管理することになる。

【0005】つまり、USBデバイス機器間通信を行う場合にPCのようなUSBホスト機器を特別に用意する必要がなくなり、ユーザはホスト機能を持ったUSBデバイス機器と、ターゲットであるUSBデバイス機器を用意するだけよい。

【0006】図12に、このホスト機能を搭載したUSBデバイス機器のブロック構成図を示している。USBデバイス機器11は、MPU111と、USBデバイスとしての機能を成すUSBデバイスコントローラ112とUSBデバイスドライバ113に加えて、USBホストとしての機能を実現するためのUSBホストコントローラ114とUSBホストドライバ115を持つ構成となる。そして、このUSBデバイス機器11とターゲットデバイスとなるUSBデバイス機器12が接続され、更にUSBホスト機器10の接続を可能とする。

【0007】このシステムで使用されるUSBケーブルは、USB規格に促して、USBデバイス機器11とUSBデバイス機器12との間の接続はコネクタタイプがAタイプとBタイプの接続となり、USBデバイス機器11とUSBホスト機器10との接続はBタイプとAタイプとの接続となる。そのため、USBデバイス機器11内の基板に用意するレセプタクル(USBケーブルのコネクタ挿入口)としては、USBデバイスとして機能する際のBタイプと、USBホストして機能する際のAタイプの2種類を用意しなければならぬコネクタ2個分のスペースが必要となる。そして、どちらのレセプタクルにUSBケーブルが接続されたかによって、そのUSBデバイス機器の動作が変わってくる。

【0008】USBデバイス機器11がUSBホスト機

能として動作する場合は、AタイプレセプタクルにUSBケーブルが接続された場合であり、その先にはターゲットとなるUSBデバイス機器12が接続される。その際に行われるUSB通信は、一般的なUSBホスト機器としての機能がそのまま備わっているため、一般的なUSBホスト機器と同じ通信方式で行われる。

【0009】

【発明が解決しようとする課題】従来のUSBデバイス機器間の通信システムでは、USBホスト機能を持たせたUSBデバイス機器に使用されるUSBケーブルは、その両端コネクタにはUSBホスト機器側へ接続するAタイプと、USBデバイス機器側へ接続するBタイプを持つケーブルが必要であった。このため、USBホスト機能を搭載するUSBデバイス機器に装備するレセプタクル(コネクタ挿入口)は、USBデバイスとして標準装備されるBタイプだけでなく、USBホストとして機能する際のAタイプを用意しなければならないという問題があった。

【0010】また、USBホスト機能を持たせたUSBデバイス機器には、USBホスト機能(USBホストコントローラとUSBホストドライバなど)をUSBデバイス機器へ実装する必要があるため、USBデバイス機器を開発するためのコストが高くなり、更に基板面積の拡大等が起こるという問題があった。

【0011】この発明は、上記の問題点を解決するためになされたもので、ハードウェア構成は通常のUSBデバイス機器とほぼ同じ構成で、USBホストコントローラ等を追加搭載する場合ほどの開発コスト、基板面積は掛からず、トーンパケット発行等の一部機能を付加するだけで、USBホスト機器を必要とせずにUSBデバイス機器同士のみでの直接データ転送を可能にするUSBデバイス機器、USBデバイス機器間の通信システム、および通信方法を提供することを目的としている。

【0012】

【課題を解決するための手段】上記目的を達成するためには本発明のUSBデバイス機器は、少なくともMPUと、USBデバイスコントローラと、USBデバイスドライバとを有するUSBデバイス機器であって、接続ケーブルによって接続される他のUSBデバイス機器を認識する処理とUSBデバイス間データ転送を実行する処理を持った簡易版ホストドライバを設け、前記簡易版ホストドライバによって前記他のUSBデバイス機器との直接データ転送を可能にしたことを特徴とする。

【0013】また、本発明のUSBデバイス機器間の通信システムは、少なくともMPUと、USBデバイスコントローラと、USBデバイスドライバとを有する第1および第2のUSBデバイス機器と、前記第1および第2のUSBデバイス機器を接続する両端Bタイプコネクタを持つケーブルと、前記第1のUSBデバイス機器に設けられた、前記第2のUSBデバイス機器を認識する

処理とUSBデバイス間データ転送を実行する処理を持った簡易版ホストドライバとを有し、前記第1のUSBデバイス機器と前記第2のUSBデバイス機器との間で直接データ転送を可能にしたことを特徴とする。

【0014】また、本発明のUSBデバイス機器間の通信方法は、少なくともMPUと、USBデバイスコントローラと、USBデバイスドライバとを有する第1および第2のUSBデバイス機器間の通信方法であって、前記第1のUSBデバイス機器に設けられた、前記第2のUSBデバイス機器を認識する処理とUSBデバイス間データ転送を実行する処理を持った簡易版ホストドライバによって、前記他のUSBデバイス機器に対し標準デバイスリクエストを必要なだけ発行して、前記他のUSBデバイス機器のデバイス情報、コンフィグレーション情報を取得し、前記他のUSBデバイスに通信情報をセットし、前記第1のUSBデバイス機器から前記第2のUSBデバイス機器へトーケンパケットのデータ通信を行なうことを特徴とする。

【0015】本発明によれば、簡易版ホストドライバをUSBデバイス機器に新たに付加することにより、USBホストとしての一部機能をソフトウェアで実現して、USBデバイス機器同士での直接データ転送を可能にする。また、ハードウェアに関して付加する機能は少なくて済むため、開発コスト、基板面積の拡大を押さえることができる。

【0016】また、USBデバイス機器間での直接データ転送を実現する際に、自明であるUSBデバイス機器の情報に関しては固定値としてホスト機能側USBデバイス機器に与えておき、USB通信時の実行処理をできるだけ省略させている。こうしてソフトウェアで実現するUSBホストとしての機能をできるだけ簡素化することにより、その簡易版ホストドライバのプログラムサイズを小さくすることができ、開発コストが低減できる。

【0017】また、本発明によれば、USBデバイス機器でありながらUSBホストとしての機能も実現可能とし、ホスト機能として動作する場合は、USB通信の際に常に通信のスタートとなるトーケンパケットの発行を行なうことが出来る。

【0018】更に、本発明によれば、USBデバイス機器間の接続をBタイプコネクタとしていることから、1つのレセプタクルだけでUSBデバイスとしてだけでなくUSBホストとしても稼動可能となることによって、2タイプ用意する場合に比べてコスト削減、基板面積縮小が出来る。

【0019】

【発明の実施の形態】以下、本発明に係る実施の形態について、図面を参照して説明する。図1乃至図2は、本発明に係るUSBデバイス機器およびUSBデバイス機器間の通信システムの第2の実施形態の構成を示すブロック図である。図1に示すように、本発明ではUSBデ

バイス機器（簡易版ホスト）20とUSBデバイス機器（ターゲットデバイス）30の間をUSBケーブル40を用いて直接接続して、データ通信を可能とする。そして、図2に示すように、USBデバイス機器（簡易版ホスト）20には、従来のUSBデバイス機器と同様にMPU201、デバイスコントローラ202、デバイスドライバ203を有すると共に、本発明を実現するために簡易版ホストドライバ204が新たに設けられる。図3は、図2のシステムにUSBホスト機器50を接続可能にした通信システムの第2の実施形態の構成図を示す。この場合、USBデバイス機器20とUSBホスト機器50の接続はコネクタタイプB-Aの接続となる。

【0020】また、USBデバイス機器20とUSBデバイス機器30の間を直接接続するUSBケーブル40のコネクタ形状を図4に示す。このシステムで使用するUSBケーブル40には、両端ともにBタイプのコネクタを持つものを新たに提供している。即ち、本発明では、両端コネクタタイプともにBタイプであるUSBケーブル40を使用することによって、USBホストとして機能する際にもUSBデバイスとして機能する場合と同様のBタイプレセプタクルを利用可能としている。これによって、USBデバイス機器（簡易版ホスト）20に装備するレセプタクルを1つで済ますことができ、コスト削減、基板面積の縮小化が図れる。

【0021】また本発明のシステムでは、従来のホストコントローラをそのままUSBデバイス機器（簡易版ホスト）20に搭載するのではなく、ハードウェアとしてはトーケンパケットの発行、通信エラー検出等の一部機能だけを付加し、USBホストとしての機能は、簡易版ホストドライバ（ソフトウェア）204によって実現する。この、簡易版ホストドライバ204によって提供される機能は少なくとも以下の2つである。

【0022】第1は、簡易エニュレーション機能（USBホスト機能として動作開始するにあたり、接続されたUSBデバイス機器を認識するための処理）である。第2は、簡易データ転送機能（目的のUSBデバイス間データ転送を実現するための処理）である。

【0023】次に、本発明によるUSBデータ通信動作の処理手順について説明する。図5、図6は、簡易版ホストを実現するUSBデバイス機器20の動作処理のフローチャートを示す。

【0024】図5は、USBデバイス機器20の電源が投入されてからUSBケーブル40が接続され、簡易版ホスト機能とUSBデバイス機能のどちらを動作させるかを判別するまでの動作を示す。この時のハードウェア条件として、以下の項目が準備される。

- (1) 簡易版ホストのUSBデバイス機器は自己電源を投入する。
- (2) 接続されたUSBケーブルを介して、USBデバイス機器30に電源を供給する。

(3) USBケーブルの挿入を検知してMPUへ通知する。

【0025】図5において、簡易版USBホストドライバ204を搭載するUSBデバイス機器20は、自己電源を投入後(ステップS10)、USBケーブル40による接続を行ない(ステップS11)、ケーブル接続検知を行なう(ステップS12)。この接続検知は、ケーブル40によってUSBデバイス機器(簡易版ホスト)20とUSBデバイス機器(ターゲットデバイス)30とが接続後、暫くしてもエニュメレーションが開始されなければ(即ち、USBデバイス機器30からSETUPトーカンパケットが送られてこなければ)、接続先がUSBデバイス機器30であると判断し、接続ケーブル40の電源ラインにUSBバス電源供給を行ない(ステップS13)、USBデバイス機器20は簡易版ホストドライバ204の動作を開始させる(ステップS14)。一方、エニュメレーション動作を開始すれば、接続先がUSBホスト機器(図3の50)であると判断して、USBデバイス機器20はデバイスドライバ203をスタートさせる(ステップS15)。

【0026】なお、USBケーブル40の接続先がUSBホスト機器かUSBデバイス機器かを判定する機能は、ハードウェア手段を用いても構わない。この場合、例えばUSBケーブル40のコネクタ、またはレセプタクルの機構自体に接続先が何かを検出する回路を設けることにより実現できる。

【0027】図6は、簡易版ホストドライバ204の動作手順を示している。図6に示したフローチャートの、ステップS20からS25のSET CONFIGURATIONまでの標準デバイスリクエストの発行と処理が「簡易エニュメレーション処理」であり、その後に実行するUSBクラスリクエスト処理が「簡易データ転送処理」である。この動作を行なう為に、ハードウェア条件として、以下の項目が準備される。

(1) トーカンパケット等、USBデバイス機器(簡易版ホスト)20からUSB通信の際に発行すべきパケット群を発行する。

(2) データ通信の際のエラー検出、リトライ等をサポートする。

【0028】本発明において、USBデバイス機器間通信が1対1であること、ターゲットとなるUSBデバイスクラスやUSB通信方式等は事前に明らかであることなどから、目的のUSB通信を実行する環境は限られた状況にある。この為、従来のUSBホスト機器の機能を全て備える必要はない。そこで、本発明では、ホスト機能をソフトウェアで実装する際に、このシステムにおいて必要不可欠なホスト機能を備えたものが簡易版ホストドライバ204である。そして、簡易版ホストドライバは、以下のような方式および手順で実現する。

【0029】(a) 簡易エニュメレーション機能 (US

B標準デバイスリクエスト処理)

ここでのエニュメレーションとは、USBデバイス機器(簡易版ホスト)20が接続されたUSBデバイス機器30を認識するために実行する処理をいう。この処理を完了することによって初めて、USBデバイス機器20が持っている通信機能を用いて簡易版ホストして動作を開始する。

【0030】エニュメレーション中では、一般的に標準デバイスリクエストという「あるデバイス情報の送受信要求」がUSBホスト機器からUSBデバイス機器へ発行される。それを受けたUSBデバイス機器は、その要求に応じたデータをUSBホスト機器へ返したり、USBデバイス機器内部の構成をセットしたりする。つまり、USBホスト機器20は標準デバイスリクエストを必要なだけ発行し、USBデバイス機器30からデバイス情報を取得したり、USBデバイス機器30のデバイス構成をセットしたりする。

【0031】このエニュメレーションを実行するために少なくとも必要とする標準デバイスリクエストのコマンド種類は以下の通りである。図6に示すように、簡易版ホストドライバ204は、これらの要求コマンドをUSBデバイス機器30へ発行し、その内容に応じた処理を行う。

【0032】GET_DESCRIPTOR(デバイス)のコマンド発行(ステップS20)。このコマンドを受信したUSBデバイス機器30は、自装置のデバイス情報をUSBデバイス機器20へ返す。通常USBクラスとして、プリンティングデバイスクラス、ヒューマンインターフェースクラス、オーディオクラス、マスストレージクラスが存在するが、上記デバイス情報によりUSBデバイス機器20は、USBデバイス機器30のUSBデバイスクラス(デバイス情報より)を取得することが出来、また、デバイス情報として持つコンフィグレーション数の取得することが出来る。

【0033】GET_DESCRIPTOR(コンフィグレーション)の発行(ステップS21)。このコマンドを受信したUSBデバイス機器30は、自機器のコンフィグレーション情報をUSBデバイス機器20へ返す。これにより、USBデバイス機器20はUSBデバイス機器30のデバイス情報として、そのコンフィグが持つインターフェース数を取得、そのインターフェースが持つエンドポイント数(デバイスクラスによってエンドポイント数が定められている)を取得、またインターフェース情報よりUSBデバイスクラスを取得することが出来る。そして、USBデバイス機器20はUSBデバイス機器30が目的のUSBデバイスクラスであるとの確認、目的のUSBデータ転送方式を行うエンドポイントナンバーの確認、目的のUSBデータ転送方式を行うエンドポイントの最大パケットサイズ確認を行なう。

【0034】上述したコマンドの発行と処理により、タ

ターゲットデバイスであるUSBデバイス機器30のUSBクラス（この例では、プリントイングデバイスクラス）、データ転送エンドポイント（EP）を確認することが出来る（ステップS22, S23）。ステップS22, S23において、ターゲットデバイスでないと判断された場合には処理を終了する。

【0035】次に、SET_ADDRESSのコマンド発行（ステップS24）。このコマンドは、USBデバイス機器20からUSBデバイス機器30を管理するために割り付けるデバイスアドレスをUSBデバイス機器30へ送り、USBデバイス機器30はその値にセットするものであるが、本発明ではUSBデバイスリクエスト中のデバイスアドレス値（wValueの内容）には事前に決定してある固定値を入れる。本発明では、予めターゲットデバイスが分かった状態で、1対1のデータ転送を前提にして行われるものであることから、そのターゲットデバイスに応じた値をセットすることにより、無駄な交信を省略することが可能となる。

【0036】次に、SET_CONFIGURATIONのコマンド発行（ステップS25）。USBデバイス機器間の構成要素（通信方式等）を決定する。これについても、USBデバイスリクエスト中（wValueの内容）に、GET_DESCRIPTORリクエストで取得した、目的のUSBデータ転送方式を実行できる構成を持つコンフィグレーション値を入れてUSBデバイス機器30へ送信する。

【0037】なお、以下に挙げたパラメータ（a）乃至（c）は、従来のUSBホストであればエニュメレーション中に取得したUSBデバイス情報などに基づいて決定するものであるが、本発明では、限られた環境における通信であることから、事前に固定値として周知のものとしておくことができ、それらに関するエニュメレーション中の処理を省略することが可能となる。

（a）ターゲット（この場合、USBデバイス機器30）とUSBコントロール転送を行うエンドポイント0の最大パケットサイズ。

（b）ターゲットの消費する最大バス電力消費量。

（c）ターゲットにセットするデバイスアドレス値。

【0038】図7には、標準デバイスリクエスト処理の際に、簡易版ホストドライバ204によって動作するUSBデバイス機器20とターゲットデバイスであるUSBデバイス機器30との間で行われる、上述した各種コマンドのUSBパケットのやり取りが示されている。USBはパケット通信であり、通信の基本はトーカンパケット（SETUP/IN/OUT）→データパケット→ハンドシェークパケットの1トランザクションである。このトーカンパケット（SETUP/OUT/IN）を発行するのは常にUSB通信の主導権を持っているUSBデバイス機器20である。

【0039】（2）簡易データ転送機能（USBクラス

リクエスト処理）

このシステムで達成すべき目的のUSBデータ通信を実行するのが、このUSBクラスリクエスト処理の実行である。USBデバイス機器20の簡易版ホストドライバ204は、ターゲットとするUSBデバイス機器30の属するUSBデバイスのクラスリクエストをサポートし、必要なクラスリクエストを発行してその内容に応じた一連の処理を行なうことによってデータ転送を実現する。

【0040】図7に示すように、先ずデバイスリクエスト処理は、セットアップステージ、データステージ、ステータスステージの順に通信処理が進められる。このデータ転送は通信エンドポイント0の通信パケットにて処理される。セットアップステージでは、USBデバイス機器20からデバイスリクエストとしてSETUPおよびDATAをUSBデバイス機器30に送信し、USBデバイス機器30はそれを受信したならばACK信号をUSBデバイス機器20に返す。

【0041】次のデータステージでは、USBデバイス機器20からデバイス情報を読み取るためにINをUSBデバイス機器30に送信し、USBデバイス機器30はそれに対しデバイス情報DATAをUSBデバイス機器20に返す。USBデバイス機器20はデバイス情報DATAを受信するとACK信号をUSBデバイス機器30に返すと共に、次のデバイス情報を読み取る為にINをUSBデバイス機器30に送信して、同様にデバイス情報の読み取り処理が行われる。

【0042】次のステータスステージでは、USBデバイス機器20からUSBデバイス機器30の設定情報を送信するために、OUTと設定情報DATAをUSBデバイス機器30に送信し、USBデバイス機器30はそれを受信したならばACK信号をUSBデバイス機器20に返す。

【0043】次に、図8を参照して、USBクラスリクエスト処理である簡易版ホストドライバ204を持つUSBデバイス機器20からターゲットデバイスであるUSBデバイス機器30へのデータ送信を実行する場合の処理手順を説明する。

【0044】本発明では、対応するUSBクラスリクエストの種類は、USBデバイス機器30として対応すべき全てのものではなく、目的のUSBデバイス機器間でのデータ転送を実現するために必要なものとしている。簡易版ホストドライバ204は、必要であればこのデータ転送の前に行うアプリケーション（データ形式変換等）のサポートも行なうようにする。

【0045】図8（a）はクラスリクエスト処理のコントロール転送を示し、（b）はパルク転送を示す。

（a）コントロール転送：コントロール転送は、図7に示した標準デバイスリクエストと同じである。そして、セットアップステージでは、SETUPトーカンパケッ

トに統いて、USBクラスリクエストのデータパケットDATAをUSBデバイス機器30へ発行する。このリクエストの種類によって、その後どのようなクラスリクエスト処理およびデータ転送を実行していくかをUSBデバイス機器30へ伝える。その際に、必要な実行コマンド(パラメータ)などはこの転送中のデータステージで送信する。

【0046】(b) バルク転送：(a)に統いて、(a)の転送中にUSBデバイス機器30へ伝えたクラスリクエスト内容の実行コマンドに従って、データ転送を実行する。通信を行う際に必要となるエンドポイント情報等は簡易エニュメレーション時に得た情報を利用するものとする。これにより、実施形態に示すUSBデバイス機器(デジタルスチルカメラ)20から画像データがUSBデバイス機器20(プリンタ)に転送され、USBデバイス機器間でのデータ通信により画像データの印刷が実行される。

【0047】上述したシステムでは、USBデバイス機器同士の直接データ転送であったが、この方式はUSBホスト機器同士の直接データ転送を可能にするシステム(第3の実施形態)にも応用できる。

【0048】即ち、図9に示すようにUSBホスト機器(50、51)同士でのUSB転送を可能とする。この場合、USBケーブル40はAタイプ-Aタイプのものを用意する。そして、図10に示すように、簡易版デバイスとなるUSBホスト機器50を構成し、USBホスト機器50の中には簡易版USBデバイスドライバ504を搭載する。コントローラはUSBホストコントローラ502のままで、USBデバイスとしてのパケットを発行する機能などをハードウェアに付加するだけであり、接続されたUSBホスト機器(ターゲットホスト)51とのデータ通信は主に簡易版USBデバイスドライバ504で実行することによって、開発コスト/基板面積の拡大を防ぎつつUSBホスト間通信を実現することが出来る。

【0049】

【発明の効果】以上、本発明によれば、USBデバイス機器間のデータ通信を達成する簡易版USBホストドライバをUSBデバイス機器に実装することによって、USBデバイス機器間を直接接続した通信システムを実現することが出来る。その実現に当たっては、使用するコントローラはUSBデバイスコントローラのみのままであり、USBデバイス機器にUSBホストコントローラを搭載する場合ほどの大幅な基板改良は必要なく、開発

コスト、基板面積の大規模な拡大は起こすことはない。また、本発明では、USBデバイス機器同士を直接接続する際に使用するUSBケーブルとして、両端ともにBタイプコネクタを持つケーブルとすることによって、簡易版ホストとして機能する際にもUSBデバイスとして機能する場合と同様のBタイプセレクタクルを利用することができるようになり、開発するUSBデバイス機器に用意するセレクタクルは1種類ですむため開発コストの低下、基板面積の縮小化につながる。

【図面の簡単な説明】

【図1】本発明のUSB通信システムの第1の実施形態の接続構成図。

【図2】本発明のUSBデバイス機器の内部構成を示すブロック図。

【図3】本発明のUSB通信システムの第2の実施形態の接続構成図。

【図4】本発明に用いられるUSBケーブルコネクタの形状を示す図。

【図5】本発明の簡易版ホストドライバ搭載のUSBデバイス機器の動作を示すフローチャート。

【図6】本発明の簡易版ホストドライバの処理動作を示すフローチャート。

【図7】本発明のUSBパケット通信のデバイスリクエスト通信を示す図。

【図8】本発明のUSBパケット通信のクラスリクエスト通信を示す図。

【図9】本発明のUSB通信システムの第3の実施形態の接続構成図。

【図10】本発明のUSB通信システムの第3の実施形態におけるUSBホスト機器の内部構成図。

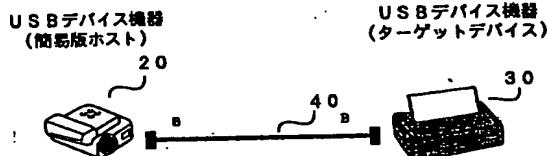
【図11】従来のUSB通信システムの構成を示す図。

【図12】従来のUSB通信システムのUSBデバイス機器の構成を示すブロック図。

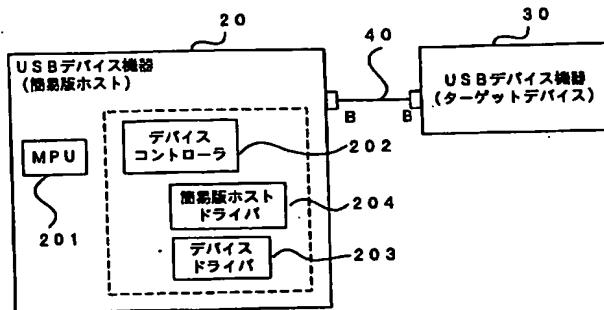
【符号の説明】

20…USBデバイス機器(簡易版ホスト)、
30…USBデバイス機器(ターゲットデバイス)
40…USBケーブル、50、51…USBホスト機器
201、501…MPU、
202…デバイスコントローラ
203…デバイスドライバ、204…簡易版ホストドライバ
502…ホストコントローラ、503…ホストドライバ
504…簡易版デバイスドライバ

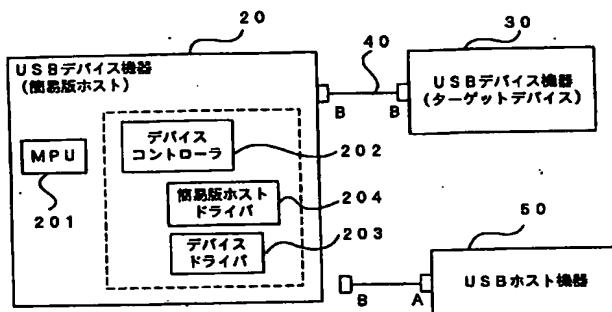
【図1】



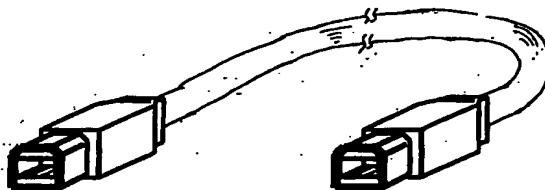
【図2】



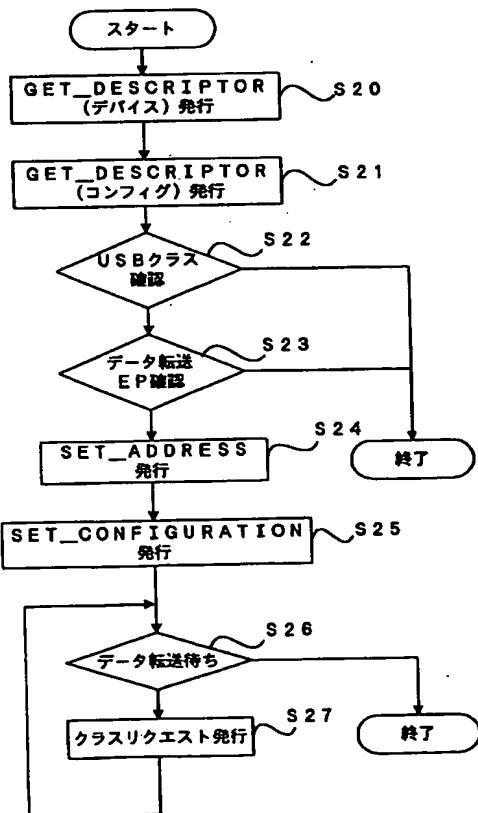
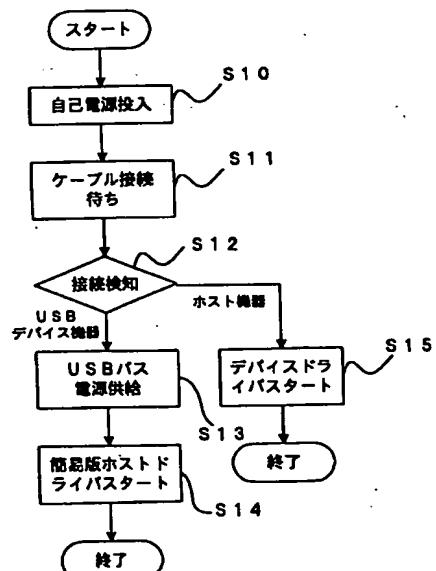
【図3】



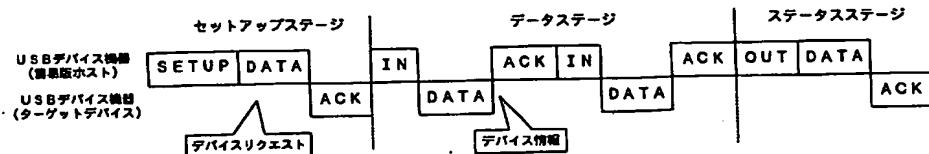
【図4】



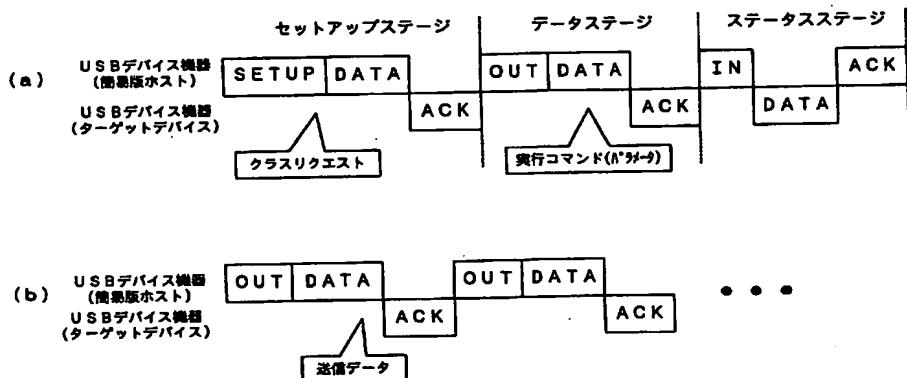
【図5】



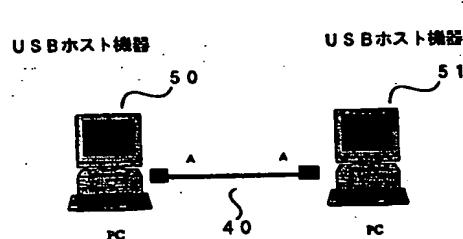
【図7】



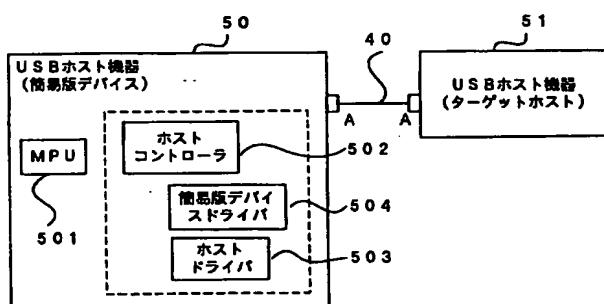
【図8】



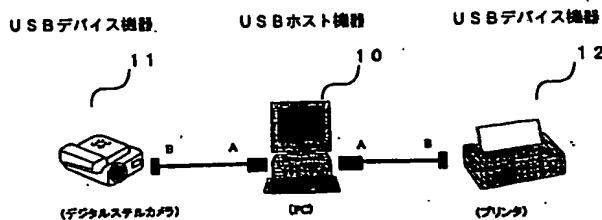
【図9】



【図10】



【図11】



【図12】

